



### Priebeh celoštátneho kola

Celoštátne kolo 39. ročníka Olympiády v informatike, kategórie A, sa koná v dňoch 20.-23. 3. 2024. Na riešenie úloh prvého, teoretického dňa majú súťažiaci 4,5 hodiny čistého času. Rôzne úlohy riešia súťažiaci na samostatné listy papiera. Akékoľvek pomôcky okrem písacích potrieb (napr. knihy, výpisy programov, kalkulačky) sú zakázané.

### Čo má obsahovať riešenie úlohy?

- Slovné popíšte algoritmus. Slovný popis riešenia musí byť jasný a zrozumiteľný i bez nahliadnutia do samotného algoritmu/programu.
- Zdôvodnite správnosť vášho algoritmu.
- Uveďte a zdôvodnite jeho časovú a pamäťovú zložitosť.
- Podrobne uveďte dôležité časti algoritmu, ideálne vo forme programu v nejakom bežnom programovacom jazyku.
- V prípade, že používate vo svojom programovacom jazyku knižnice, ktoré obsahujú implementované dátové štruktúry a algoritmy (napr. STL pre C++), v popise algoritmu stručne vysvetlite, ako by ste napísali program s rovnakou časovou zložitou bez použitia knižnice.

### Hodnotenie riešení prvého (teoretického) dňa

Za každú úlohu môžete získať od 0 do 10 bodov.

Pokiaľ nie je v zadaní povedané ináč, najdôležitejšie dve kritériá hodnotenia sú v prvom rade **správnosť** a v druhom rade **efektívnosť** navrhnutého algoritmu. Na výslednom počte bodov sa môže prejaviť aj kvalita popisu riešenia a zdôvodnenie tvrdení o jeho správnosti a efektívnosti.

Efektívnosť algoritmu posudzujeme vypočítaním jeho časovej zložitosti – funkcie, ktorá hovorí, ako dlho vykonanie algoritmu trvá v závislosti od veľkosti vstupných parametrov. Nezávisí pri tom na konštantných faktoroch, len na rádovej rýchlosti rastu tejto funkcie.

V zadaní úlohy môžu byť uvedené limity na veľkosť premenných. Tieto môžete použiť na odhad toho, ako dobré vaše riešenie je. Na počítači, ktorý vykoná miliardu inštrukcií za sekundu, vyrieši vzorové riešenie ľubovoľný povolený vstup nanaajvyš za niekoľko sekúnd.



### A-III-1 Suši bufet

V suši bufete je pohyblivý pás. Po páse dnes postupne išlo  $n$  tanierikov. Na  $i$ -tom tanieriku bol kúsok suši, ktorý mal kvalitu  $q_i$  a vážil  $v_i$  gramov.

Vieme, že niekedy počas dňa prišla do bufetu Kika. Keď tam prišla, sadla si ku pásu, nejaký čas jedla za radom všetky suši, ktoré šli okolo, a keď už usúdila, že toho zjedla dosť, tak zase z bufetu odišla.

Samozrejme, keď zjeme desať rôznych suši a jedno z nich je zlé, pokazí to dojem z celého obeda. Preto platí, že kvalita Kikinho obeda je rovná **najmenejšej** spomedzi kvalít suši, ktoré zjedla.

#### Súťažná úloha

Vieme, že Kika sa v bufete zvládla celkom dosť najesť. Presnejšie, vieme, že dokopy zjedla aspoň  $z$  gramov suši.

Zistite, akú najväčšiu kvalitu  $q_{max}$  mohol mať Kikin obed. Následne zistite, koľko najviac gramov ( $v_{max}$ ) suši mohla zjesť, ak mal jej obed kvalitu  $q_{max}$ .

#### Formát vstupu a výstupu

V prvom riadku sú celé čísla  $n$  a  $z$ .

V druhom sú nezáporné celé čísla  $q_1, \dots, q_n$ : kvality jednotlivých suši v poradí, v ktorom išli po páse.

V treťom riadku sú nezáporné celé čísla  $v_1, \dots, v_n$ : váhy jednotlivých suši v tom istom poradí.

Na výstup vypíšete dva riadky: v prvom číslo  $q_{max}$ , v druhom číslo  $v_{max}$ .

#### Obmedzenia a hodnotenie

Pri písaní programu predpokladajte, že aj celková váha všetkých suši aj ich najväčšia kvalita sa zmestia do bežnej celočíselnej premennej. Tieto čísla však môžu byť rádovo väčšie ako  $n$ , a teda ich napr. nevieme usporiadať v čase lineárnom od  $n$ .

Plný počet bodov môžu získať riešenia s časovou zložitostou lineárnou od  $n$ .

Nanajvýš 8 bodov môžu získať riešenia s časovou zložitostou  $O(n \log n)$ .

Nanajvýš 4 body môžu získať riešenia s časovou zložitostou kvadratickou od  $n$ .

Pomalšie riešenia dostanú ešte menej bodov. Riešenia, ktorých časová zložitosť je medzi 4-bodovým a 8-bodovým riešením, dostanú od 5 do 7 bodov v závislosti od ich zložitosti.

#### Príklady

vstup

```
6 140
400 300 100 400 1000 10
50 51 52 53 54 20
```

výstup

```
100
260
```

*Kika určite zjedla tretie suši, jej kvalita obeda bola teda nanajvýš 100. Najväčší možný obed s touto kvalitou sa skladá z prvých piatich suši.*

vstup

```
6 107
400 300 100 400 1000 10
50 51 52 53 54 20
```

výstup

```
400
107
```

*Keď Kika zje štvrté a piate suši, jej obed bude mať kvalitu  $\min(400, 1000) = 400$  a celkovú hmotnosť presne  $53 + 54 = 107$  gramov.*



### A-III-2 Pexeso

Na dlhom pásiku papiera je za sebou nakreslených  $n$  rovnako veľkých štvorcových obrázkov. Obrázky na pásiku sa môžu ľubovoľne opakovať.

Kubík si chce vystrihnúť kúsok tohto pásika (t.j. súvislý úsek tvorený niekoľkými po sebe idúcimi obrázkami) a vyrobiť si z neho pexeso. Ako určíte viete, v pexese chceme mať každý obrázok presne dvakrát.

Kubík sa rozhodol, že do svojho pexesa dá práve tie obrázky, ktoré budú na jeho kúsku **presne dvakrát**. (Teoreticky by vedel použiť aj tie obrázky, ktoré má na svojom kúsku viackrát, ale na to by musel z každého zahodiť správny počet nadbytočných kópií a s tým je moc oštary.)

#### Súťažná úloha

Na vstupe je daný popis pásika: postupnosť čísel  $a_0, \dots, a_{n-1}$ , pričom rovnaké čísla predstavujú rovnaký obrázok. Napíšte program, ktorý vypočíta, koľko najviac dvojíc môže mať Kubíkovo pexeso.

#### Formát vstupu a výstupu

V prvom riadku vstupu je celé číslo  $n$ , v druhom sú celé čísla  $a_0, \dots, a_{n-1}$ .

Na výstup vypíšete jedno číslo: najväčší možný počet dvojíc na súvislom kúsku papiera.

#### Obmedzenia a hodnotenie

O veľkostiach čísel  $a_i$  nepredpokladajte nič iné ako to, že sa zmestia do bežných celočíselných premenných.

Plný počet bodov môžu získať riešenia s časovou zložitostou  $O(n \log n)$ .

Za každé riešenie s lepšou ako kvadratickou časovou zložitostou sa dá získať aspoň 8 bodov.

Nanajvýš 5 bodov dostanú riešenia, ktoré majú **aj v najhoršom prípade** časovú zložitost kvadratickú od  $n$ .

Nanajvýš 4 body môžu získať iné riešenia, ktorých časová zložitost je približne kvadratická od  $n$ .

Pomalšie riešenia dostanú ešte menej bodov.

#### Príklady

vstup

```
14
7 2 2 1 7 7 999 7 999 7 2 1 1 7
```

výstup

```
3
```

Ak si Kubík vystrihne kúsok „2 1 7 7 999 7 999 7 2 1“, môže si dať do pexesa dvojice s obrázkami 1, 2 a 999. Iným optimálnym riešením je vystrihnúť kúsok „999 7 999 7 2 1 1“ a do pexesa zobrať obrázky 999, 7 a 1. Žiaden kúsok tohto papiera neobsahuje viac ako tri dvojice.

vstup

```
9
10 20 30 40 50 40 30 20 10
```

výstup

```
4
```

Tu je pre Kubíka najlepšou možnosťou zobrať úplne celý pásik papiera.

vstup

```
9
10 10 10 20 20 20 30 30 30
```

výstup

```
2
```

Tu je optimálne vystrihnúť buď kúsok „10 10 20 20“, kúsok „20 20 30 30“, alebo kúsok „10 10 20 20 20 30 30“. Z každého z nich dostane Kubík pexeso s dvoma dvojicami.



### A-III-3 O Vekslákbotovi a Pokladničke

K tejto úlohe patrí študijný text uvedený nižšie. Je identický so študijným textom v zadaniach domáceho kola. Každá podúloha je hodnotená samostatne. Nezabudnite uvádzať aj **slovný popis** svojich programov.

#### Podúloha A (2 body): majorita

V Pokladničke je na začiatku  $c > 0$  červených,  $m > 0$  modrých a  $z > 0$  zelených žetónov (a nič iné). Je zaručené, že **niektorá farba má majoritu** – teda žetónov danej farby je *ostro viac* ako polovica všetkých žetónov. Napíšte program, po skončení ktorého bude v Pokladničke presne jeden žetón, a to tej farby, ktorá mala pôvodne majoritu.

#### Podúloha B (4 body): logaritmus

Funkcia  $g(x) = \lceil \log_2 x \rceil$  počíta hornú celú časť binárneho logaritmu čísla  $x$ . Slovné:  $g(x)$  je najmenšie celé číslo  $y$  také, že  $2^y \geq x$ . Napríklad platí  $g(31) = 5$ ,  $g(32) = 5$  (lebo  $2^5$  je presne 32), ale potom už  $g(33) = 6$ .

Na začiatku je v Pokladničke  $c > 0$  červených žetónov a nič iné. Napíšte program, po skončení ktorého bude v Pokladničke presne  $g(c)$  modrých žetónov. Na konci môžu ostať v Pokladničke aj žetóny iných farieb.

#### Podúloha C (4 body): Fibonacci

Fibonacciho čísla sú definované nasledovne:  $F_0 = 0$ ,  $F_1 = 1$  a pre všetky  $n \geq 2$  platí  $F_n = F_{n-1} + F_{n-2}$ . Táto postupnosť teda začína nasledovne: 0, 1, 1, 2, 3, 5, 8, 13, 21, ...

V Pokladničke je na začiatku  $c > 0$  červených žetónov a nič iné. Napíšte program **bez použitia obmedzení**, po skončení ktorého bude v Pokladničke presne  $F_c$  modrých žetónov a *nič iné*.

Čiastočné body môžete dostať za korektné riešenia, ktoré použijú obmedzenia a/alebo budú mať na konci v Pokladničke aj iné ako modré žetóny.

## Študijný text: O Vekslákbotovi a Pokladničke

Za 111 horami a 111 dolinami leží jedna prazvláštna krajina. V tejto krajine žijú samí roboti a ako platidlá používajú žetóny všetkých možných farieb od výmyslu sveta. V domčeku na úpätí ďalšej hory si tam spolu nažívajú dvaja hrdinovia nášho príbehu: roboti Vekslákbót a Pokladnička.

Ako možno tušíte z jej mena, Pokladnička v sebe rada uskladňuje všetky možné žetóny. Ako zrejme netušíte z jej mena, Pokladnička tiež veľmi rada vykonáva rôzne programy. A ako ste si po prečítaní predchádzajúcej vety úplne istí (koniec koncov, toto je študijný text Olympiády v informatike a nie len taká rozprávka), práve tieto programy pre ňu budete písať vy ako riešenia súťažných úloh.

Vekslákbót je majster vo vymieňaní žetónov: či už chcete vymeniť dva červené za tri modré alebo žltý a čierny za miliónpäť sivých, Vekslákbót určite pozná robota, ktorý pozná robota, ktorý s ním práve takú výmenu rád spraví. Vekslákbót má veľmi rád Pokladničku, a tak keď ho ona o hocijakú výmenu poprosí, okamžite (alebo, ako hovoríme my, v konštantnom čase) jej ju zabezpečí.

Poďme sa teda pozrieť na to, ako vlastne budú vyzeráť programy pre našu Pokladničku.

#### Základy: vstup, výstup, program

„Vstupom“ pre Pokladničku budú jednoducho žetóny, ktoré má na začiatku v sebe. Neexistuje nijaký výstup. V zadaní rôznych úloh budeme rôzne definovať ciele, ktoré majú vaše programy dosiahnuť.

Program pre Pokladničku sa skladá z dvoch častí: *obmedzení*, ktoré musí dodržať, a *pokynov*, ktoré má vykonávať.

#### Obmedzenia

Prvou časťou programu pre Pokladničku je **konečná množina** obmedzení. (Táto množina môže byť aj prázdna.) Obmedzenia pre Pokladničku hovoria, najviac koľko žetónov konkrétnej farby, prípadne kombinácií žetónov rôznych farieb, smie mať naraz v sebe.

Formálne, obmedzenia sú lineárne nerovnosti nasledovného tvaru:

$$k_1 \cdot \text{farba}_1 + k_2 \cdot \text{farba}_2 + \dots + k_n \cdot \text{farba}_n \leq \text{limit}$$



pričom všetky koeficienty  $k_i$  aj limit sú konkrétne kladné celé čísla, zatiaľ čo  $\text{farba}_i$  sú premenné označujúce počet žetónov príslušnej farby v Pokladničke.

Príkladmi obmedzení sú napríklad nerovnosti „modrá  $\leq 7$ “ a „modrá + 2 · červená  $\leq 3$ “.

Ak sa nejakej farby netýka žiadne obmedzenie, môže byť v Pokladničke ľubovoľne veľa žetónov tejto farby.

### Jedna inštrukcia

Každá inštrukcia pre Pokladničku má nasledovný tvar:

$$k_1 \cdot \text{farba}_1, \dots, k_n \cdot \text{farba}_n \rightarrow \ell_1 \cdot \text{farba}_{n+1}, \dots, \ell_m \cdot \text{farba}_{n+m}$$

Všetko naľavo od  $\rightarrow$  budeme volať ľavá strana inštrukcie, všetko napravo zase pravá strana.

Všetky  $k_i$  aj  $\ell_j$  musia byť kladné celé čísla. V rámci ľavej aj v rámci pravej strany inštrukcie musia byť všetky použité farby rôzne. Je povolené použiť tú istú farbu na oboch stranách inštrukcie. Je povolené mať  $n = 0$  alebo  $m = 0$ , teda inštrukciu, ktorá má niektorú stranu prázdnu.<sup>1</sup>

Príklady inštrukcií:

- 2 červená  $\rightarrow$  3 modrá
- 1 žltá, 1 čierna  $\rightarrow$  1 000 005 sivá
- 3 červená  $\rightarrow$  333 červená, 334 cyklámenová, 335 purpurová
- 2 zelená  $\rightarrow$   $\emptyset$

Posledná inštrukcia z príkladu má prázdnu pravú stranu. Na zápis prázdnej strany inštrukcie budeme používať symbol prázdnej množiny, aby bolo jasné, že je prázdna úmyselne.<sup>2</sup>

V našich príkladoch budeme používať reálne názvy farieb. Vo svojich programoch môžete ako názvy farieb používať aj ľubovoľné iné alfanumerické reťazce. Je tiež OK pri zápise vynechať koeficient 1. Druhú z vyššie uvedených inštrukcií teda môžete zapísať aj „žltá, čierna  $\rightarrow$  1 000 005 sivá“.

Pokladnička vykoná inštrukciu tak, že zo seba vyberie sadu žetónov na ľavej strane inštrukcie, dá ich Vekslákbotovi a poprosí ho, aby jej za ne priniesol sadu žetónov z pravej strany. Vekslákbot to samozrejme promptne zabezpečí a Pokladnička do seba vloží žetóny, ktoré jej priniesol.

Ak v sebe Pokladnička nemá všetky žetóny, ktoré si vyžaduje ľavá strana inštrukcie, tak danú inštrukciu v danej chvíli vykonať nevie. Ak by napríklad mala len dva červené žetóny, nevie vykonať inštrukciu „3 červené  $\rightarrow$  333 červených“.

Pokladnička tiež nevie vykonať inštrukciu, ak by po jej vykonaní bolo porušené hociktoré z obmedzení.

### Pokyny

Pokyny pre Pokladničku tvorí **konečná postupnosť** inštrukcií vyššie uvedeného typu. Zdôrazňujeme, že ide o postupnosť, a teda **záleží** na poradí inštrukcií.

### Vykonávanie programu

Program sa vykonáva v krokoch. V každom kroku Pokladnička začne čítať pokyny od začiatku a číta ich, až kým nenájde prvú inštrukciu, ktorú vie momentálne vykonať. Tú inštrukciu vykoná. (Na zvyšok pokynov sa v tomto kroku už ani nepozrie. V ďalšom kroku začne znova čítať postupnosť inštrukcií od začiatku.)

Vykonávanie programu skončí, keď sa už žiadna inštrukcia v pokynoch nedá vykonať.

### Príklad #1: viac červených

Úloha: Na začiatku sú v Pokladničke nejaké červené a nejaké modré žetóny. Napíšte program, po ktorého skončení bude v Pokladničke práve jeden zlatý žetón a nič iné, ak bolo červených žetónov viac ako modrých. Vo všetkých ostatných prípadoch musí Pokladnička skončiť úplne prázdna.

<sup>1</sup>Povolené je mať prázdne aj obe strany inštrukcie, ale ako sa čoskoro dozvieme, program, v ktorom takúto inštrukciu použijeme, zaručene nikdy neskončí.

<sup>2</sup>Formálne môžeme povedať, že aj pravá aj ľavá strana každej inštrukcie obsahuje nejakú multimnožinu farieb. Prázdna strana inštrukcie je teda naozaj prázdna multimnožina.



### Riešenie 1: bez obmedzení

Nebudeme mať žiadne obmedzenia. Pokyny budú vyzeráť nasledovne:

1. červená, modrá  $\rightarrow \emptyset$
2. červená, zlatá  $\rightarrow$  zlatá
3. červená  $\rightarrow$  zlatá
4. modrá  $\rightarrow \emptyset$

Pri vykonávaní tohto programu bude Pokladnička najskôr používať inštrukciu 1, kým to ide. Keď to prestane ísť, má už v sebe buď len červené alebo len modré žetóny. Ak sú len modré, jediná inštrukcia, ktorá sa dá používať, je inštrukcia 4. Jej opakovaným použitím sa Pokladnička vyprázdni a program skončí.

Ak sú po skončení používania inštrukcie 1 v Pokladničke len červené žetóny, bude to o čosi komplikovanejšie. Jediná inštrukcia, ktorá sa dá v tejto situácii použiť, je inštrukcia 3: vymeníme jeden červený žetón za jeden zlatý. Od tejto chvíle sa už inštrukcia 3 nepoužije, a to preto, že už sa dá používať inštrukcia 2. Tou sa Pokladnička postupne zbaví zostávajúcich červených žetónov. Akonáhle v nej ostane len samotný zlatý žetón, už sa nedá použiť žiadna inštrukcia, a teda výpočet končí.

### Riešenie 2: s obmedzením

Budeme mať jedno obmedzenie:

- zlatá  $\leq 1$

Pokyny budú vyzeráť nasledovne:

1. červená, modrá  $\rightarrow \emptyset$
2. červená  $\rightarrow$  zlatá
3. červená  $\rightarrow \emptyset$
4. modrá  $\rightarrow \emptyset$

Tentokrát sa v situácii, kedy sú v Pokladničke len samé červené žetóny, najskôr raz vykoná inštrukcia 2 (jeden vymeníme za zlatý) a potom sa už bude používať inštrukcia 3, až kým sa všetky červené neminú. Obmedzenie, ktoré sme si zvolili, totiž bráni Pokladničke opakovane využiť inštrukciu 2.

### Príklad #2: súčet

Úloha: Na začiatku je v Pokladničke  $c > 0$  červených žetónov,  $m > 0$  modrých a jeden zelený. Napíšte program, po ktorého skončení bude v Pokladničke presne  $c$  červených,  $m$  modrých a  $c + m$  fialových žetónov.

**Riešenie.** Keby sme len chceli dostať  $c + m$  fialových žetónov, stačilo by vymeniť všetky červené aj všetky modré za fialové „s kurzom jedna k jednej“. Ako ale nestratiť pôvodné žetóny?

Naše riešenie nebude mať žiadne obmedzenia a bude mať nasledujúce inštrukcie:

1. červená, zelená  $\rightarrow$  tmavočervená, zelená
2. modrá, zelená  $\rightarrow$  tmavomodrá, zelená
3. zelená  $\rightarrow$  žltá
4. tmavočervená, žltá  $\rightarrow$  červená, fialová, žltá
5. tmavomodrá, žltá  $\rightarrow$  modrá, fialová, žltá
6. žltá  $\rightarrow \emptyset$

Všimnite si, ako náš program používa prítomnosť zeleného a žltého žetónu v Pokladničke: pomocou nich vieme rozlíšiť, či ešte premieňame pôvodné červené a modré žetóny alebo či už naspäť vznikajú nové.

## TRIDSIATY DEVIATY ROČNÍK OLYMPIÁDY V INFORMATIKE

Príprava úloh: Michal Anderle, Michal Forišek  
Recenzia: Michal Forišek, Paulína Smolárová, Truc Lam Bui  
Slovenská komisia Olympiády v informatike  
Vydal: NIVAM – Národný inštitút vzdelávania a mládeže, Bratislava 2024